DART: A FAST HEURISTIC ALGEBRAIC RECONSTRUCTION ALGORITHM FOR DISCRETE TOMOGRAPHY

K.J. Batenburg and J. Sijbers

University of Antwerp, Vision Lab
Universiteitsplein 1, B-2610, Wilrijk, Belgium

ABSTRACT

Discrete tomography (DT) is concerned with the tomographic reconstruction of images that consist of only a small number of gray levels. DT reconstruction problems are usually underdetermined. Therefore, incorporation of heuristic rules to guide the reconstruction algorithm towards an optimal as well as intuitive solution would be valuable.

In this paper, we introduce DART: a new, heuristic DT algorithm that is based on an iterative algebraic reconstruction method. Starting from a continuous reconstruction, a discrete image is reconstructed by consistent updating of border pixels. Using simulation experiments, it is shown that the DART algorithm is capable of computing high quality reconstructions from substantially fewer projections than required for conventional continuous tomography.

Index Terms— Discrete tomography, algebraic reconstruction technique, image reconstruction

1. INTRODUCTION

Discrete tomography (DT) is concerned with the problem of recovering images from their projections where the images are assumed to consist of a small number of gray values only [1]. Potential benefits of DT are an increase of the reconstruction quality and a reduction of the required number of projection images. The DT reconstruction problem, however, is generally underdetermined and the number of possible solutions can be substantial. Therefore, incorporation of additional rules to guide the reconstruction process towards an optimal as well as intuitive solution would be valuable.

Several reconstruction algorithms for DT have been proposed, most of which are limited to the reconstruction of binary images (i.e., black-and-white) [2–4].

In this paper, we propose a new, heuristic DT algorithm that is based on an iterative algebraic reconstruction algorithm. The method will be referred to as DART (Discrete Algebraic Reconstruction Technique). After introducing basic notations and concepts in Section 2.1, an overview of the DART algorithm is given in Section 2.2, which will be elaborated on in Section 2.3. Finally, in Section 3, simulation results are presented and discussed.

This work was financially supported by the the F.W.O. (Fund for Scientific Research - Flanders, Belgium)

2. METHOD

2.1. Notation and concepts

For the sake of simplicity, we assume a two-dimensional (2D) parallel projection geometry (see Figure 1), although the DART algorithm can be easily extended to other transmission tomography setups.

The *unknown object* that we want to reconstruct is represented by a gray-scale image, which is considered as a function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ with bounded support. The *projection function* $P_\theta : \mathbb{R} \rightarrow \mathbb{R}$ of f for an angle θ is defined as

$$P_{\theta,f}(t) = \iint_{-\infty}^{+\infty} f(x,y)\delta(x\cos\theta + y\sin\theta - t)\,\mathrm{d}x\,\mathrm{d}y, \quad (1)$$

with $\delta(.)$ denoting the Dirac delta function. The values $P_{\theta,f}(t)$ are often called *line projections*. The reconstruction problem consists of recovering the image f from its projection functions along a set of angles.

The reconstruction is computed on a rectangular pixel array of width w and height h. Hence, the total number of pixels in the reconstruction is given by $n = wh$. Let d be the total number of available projections, corresponding to angles $\{\theta_1, \ldots, \theta_d\}$. For each projection angle, we assume that the projection of the unknown original image was measured by an array of k equally spaced detector cells. The total number of measurement values is denoted by $m = kd$.

The reconstruction problem can be formulated as a system of linear equations (see, e.g., Chapter 7 of [5]):

$$\boldsymbol{W}\boldsymbol{x} = \boldsymbol{p}. \quad (2)$$

The $m \times n$ matrix $\boldsymbol{W}$ is called the *projection matrix*. The entries of the $n \times 1$ column vector $\boldsymbol{x}$ correspond to the pixel values of the reconstruction. The $m \times 1$ column vector $\boldsymbol{p}$ contains the measured line projections.

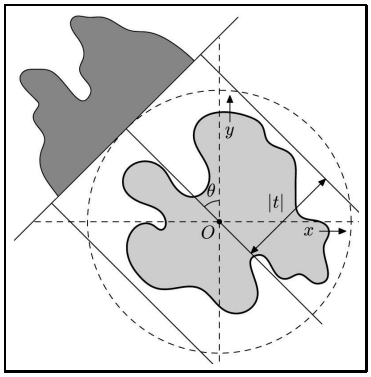


Fig. 1. Parallel projection geometry

There are a variety of algebraic reconstruction methods for continuous tomography (ART, SART, SIRT, etc.). We refer to [5] for an overview of algebraic reconstruction methods. The DART algorithm can be used in conjunction with each of these algorithms. In the context of this paper, we refer to *algebraic reconstruction method* (*ARM*) as a specific iterative reconstruction algorithm. In each ARM iteration, all projections are enumerated in random order, each time updating the current reconstruction. Define $S_q = \{1+(q-1)k, \ldots, 1+qk\}$ for $q = 1, \ldots, d$. The set S_k contains the indices of the projection matrix rows that correspond to projection q. Let $\boldsymbol{x}$ be the current reconstruction after a certain number of update steps have been performed. From $\boldsymbol{x}$, the new reconstruction $\boldsymbol{x}'$, based on projection q, is computed according to

$$x'_j = x_j + \frac{1}{\sum_{i \in S_q} w_{ij}} \sum_{i \in S_q} \frac{w_{ij}(p_i - [\boldsymbol{W}\boldsymbol{x}]_i)}{\sum_{j=1}^{n} w_{ij}} \quad ,$$

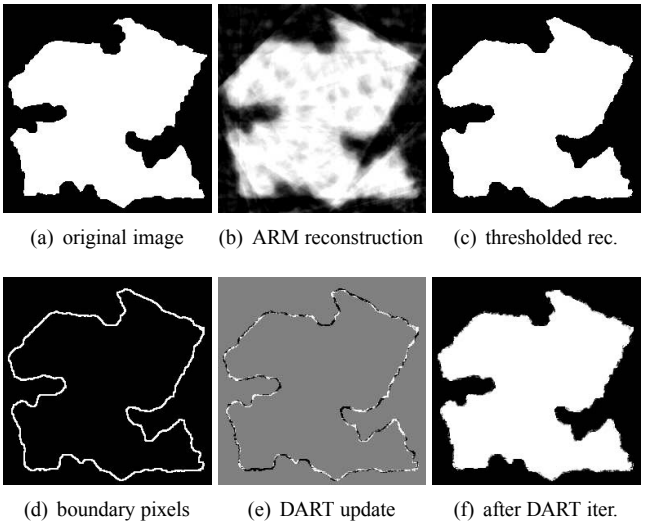
where w_{ij} is the ij^{th} element of $\boldsymbol{W}$. We use the term *ARM iteration* to denote a sequence of d such update steps, applied for each of the projections in random order.

Besides the projection data, the DART algorithm requires as input the set of gray levels in the reconstruction, which we assume to be known in advance. If ℓ is the number of gray levels in the image, then the set of gray levels will be denoted by by $R = \{\rho_1, \ldots, \rho_\ell\}$.

2.2. Algorithm overview

Before giving a concise description of the operations performed by DART, we will first give a brief overview of the algorithmic ideas.

Suppose that we want to reconstruct the binary image shown in Figure 2(a) from only 12 projections. Figure 2(b) shows the ARM reconstruction after ten iterations. From the ARM reconstruction, it is difficult to decide where the edges of the object are exactly. Yet, the thresholded reconstruction in Figure 2(c) shows that pixels of the interior of the object that are not too close to the boundary are assigned the correct gray level in the thresholded image. The same holds for pixels



(a) original image (b) ARM reconstruction (c) thresholded rec.

(d) boundary pixels (e) DART update (f) after DART iter.

Fig. 2. Reconstructing a phantom. The images indicate the various steps of the DART algorithm.

in the background region that are far away from the object boundary. Let B be the boundary of the object in the thresholded image, which is defined as the set of all pixels that are adjacent to at least one pixel having a different gray level (cfr. Figure 2(d)). We now move back to the original gray level ARM reconstruction. All pixels that are not in B are assigned their thresholded value, either black or white. Next, several ARM iterations are performed for the pixels in B only. In this way, we significantly reduce the number of variables in the linear equation system (2), while the number of equations remains the same. Figure 2(e) shows the relative change of the image pixels after one ARM iteration of the boundary pixels, where gray denotes no change. In regions of the boundary B where too many white pixels were assigned the wrong gray value, the surrounding boundary pixels have strongly negative (dark) pixel values, to compensate. The opposite occurs at parts of the boundary where the extent of the background has been overestimated in the first thresholded ARM reconstruction. In this way, the values of the boundary pixels indicate how the boundary should be adapted in a new estimate of the object. In the ARM step, each of the boundary pixels is allowed to vary independently, which may result in large local variations of the pixel values. To regularize the reconstruction algorithm, the boundary pixels are locally smoothed after applying ARM. Figure 2(f) shows the result of this filtering operation. Subsequently, the resulting image is again thresholded and each of the steps that we just described is repeated iteratively.

Figure 3 shows an overview of the DART algorithm. In the next subsections, we will describe each of the steps in more detail.

Compute a start reconstruction $\boldsymbol{x}^0$ using ARM;

$t := 0$;

while (stop criterion is not met) **do**
begin

$t := t + 1$;

Compute the segmented image $\boldsymbol{s}^t = \boldsymbol{r}(\boldsymbol{x}^{t-1})$;

Compute the set I^t of non-boundary pixels of $\boldsymbol{s}^t$;

Compute the image $\boldsymbol{y}^t$ from $\boldsymbol{x}^{t-1}$ and $\boldsymbol{s}^t$, setting $y_i^t := s_i^t$ if $i \in I^t$ and $y_i^t := x_i^{t-1}$ otherwise;

Using $\boldsymbol{y}^t$ as the start solution, compute the ARM reconstruction $\boldsymbol{x}^t$, while keeping the pixels in I fixed;

Apply a smoothing operation to the pixels that are not in I;

end

Fig. 3. Basic steps of the DART algorithm.

2.3. Algorithm description

The first approximate reconstruction $\boldsymbol{x}^0$ is computed using the continuous ARM algorithm. For all DT experiments in Section 3 we used three ARM iterations.

Each time a (partially) continuous reconstruction has been computed, it is segmented to obtain an image $\boldsymbol{s}^t$ that has only gray levels from the set $R = \{\rho_1, \ldots, \rho_\ell\}$. For all experiments in this paper we used global thresholding with fixed thresholds to perform this segmentation. For $i = 1, \ldots, \ell - 1$, define

$$\tau_i = \frac{\rho_i + \rho_{i+1}}{2}.$$

Define the *threshold function* $r : \mathbb{R} \to R$ as

$$r(v) = \begin{cases} \rho_1 & (v < \tau_1) \\ \rho_2 & (\tau_1 \le v < \tau_2) \\ \cdots & \\ \rho_\ell & (\tau_{\ell-1} \le v). \end{cases} \quad . \tag{3}$$

As a shorthand notation we also define the threshold function of an *image* $\boldsymbol{x} \in \mathbb{R}^n$: $\boldsymbol{r}(\boldsymbol{x}) = \begin{pmatrix} r(x_1) & \ldots & r(x_n) \end{pmatrix}^T$.

After the segmented reconstruction $\boldsymbol{s}^t = \boldsymbol{r}(\boldsymbol{x}^{t-1})$ has been computed, the set I^t of *non-boundary pixels* is computed from the segmented image. A pixel s_i^t is called a non-boundary pixel if all pixels from its 8-connected neighborhood have the same gray level in $\boldsymbol{s}^t$. The remaining set of pixels, B^t, are called the *boundary pixels*.

Then, the image $\boldsymbol{y}^t$ is computed from $\boldsymbol{x}^{t-1}$ and $\boldsymbol{s}^t$, setting $y_i^t := s_i^t$ if $i \in I^t$ and $y_i^t := x_i^{t-1}$ if $i \in B^t$.

Consider the system of linear equations

$$\begin{pmatrix} | & & | \\ \boldsymbol{w}_1 & \cdots & \boldsymbol{w}_n \\ | & & | \end{pmatrix} \begin{pmatrix} x_1 \\ \cdots \\ x_n \end{pmatrix} = \boldsymbol{p}, \tag{4}$$

where $\boldsymbol{w}_i$ denotes the ith column vector of $\boldsymbol{W}$. We now define the operation of *fixing* a variable x_i at value $v_i \in \mathbb{R}$. It transforms the system (4) into the new system

$$\begin{pmatrix} | & & | & | & & | \\ \boldsymbol{w}_1 & \cdots & \boldsymbol{w}_{i-1} & \boldsymbol{w}_{i+1} & \cdots & \boldsymbol{w}_n \\ | & & | & | & & | \end{pmatrix} \begin{pmatrix} x_1 \\ \cdots \\ x_{i-1} \\ x_{i+1} \\ \cdots \\ x_n \end{pmatrix} = \boldsymbol{p} - v_i \boldsymbol{w}_i. \tag{5}$$

The new system has the same number of equations as the original system, whereas the number of variables is decreased by one.

Using the image $\boldsymbol{y}^t$ as the start solution, the new reconstruction $\boldsymbol{x}^t$ is computed by applying a single ARM iteration, keeping all non-boundary pixels fixed.

Finally, a local smoothening operator is applied to the pixels in the set B^t, setting $x_i^t := 0.7x_i^t + 0.3b$, where b denotes the average gray value in the 8-neighborhood of pixel i.

To determine when the algorithm should terminate, we use the *total projection error* $E : \mathbb{R}^n \to \mathbb{R}$, defined as

$$E(\boldsymbol{x}) = |\boldsymbol{W}\boldsymbol{x} - \boldsymbol{p}|_2.$$

The algorithm is terminated after the total projection error of the best reconstruction found so far has not decreased during the last 10 iterations. The termination check is computationally expensive by itself, and is therefore only performed after each multiple of 10 main loop iterations. We also used a fixed upper bound of 500 main loop iterations, which was never reached in the experiments for this paper.

3. EXPERIMENTAL RESULTS

We implemented the DART algorithm in C++, using the gcc compiler. All experiments were performed on an Intel E6700 PC, using a single CPU core.

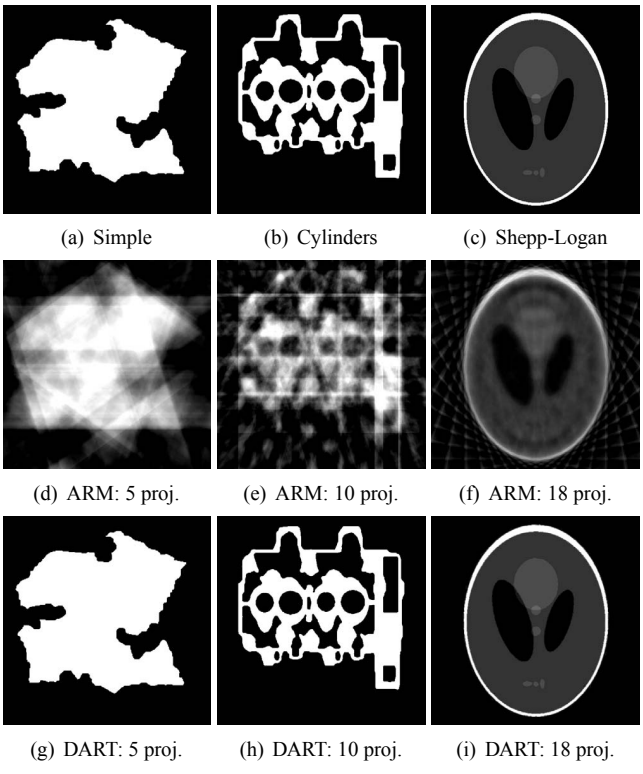
The top row of Figure 4 shows the three phantoms that were used for the reconstruction experiments. All three phantoms have size 512×512 pixels. The phantom shown in Figure 4(a), called *simple* is a relatively simple shape without any holes, although it is not convex. The second phantom, shown in Figure 4(b), called *cylinders*, represents a cross-section of a cylinder head from a motor block. It is a far more complex shape than the first phantom. The third phantom, shown in Figure 4(c) is the well-known Shepp-Logan phantom [5], which we use here to demonstrate that the DART algorithm can be used effectively when the reconstruction contains more than two gray levels.

Table 1 shows reconstruction results of the DART algorithm, using a varying number of projections. In all cases, the d projection angles are equally spaced between 0 and 180 degrees. The column "#DART iters" contains the number of times the main DART loop was executed. The column "pixel

error" indicates the fraction of pixels for which the reconstruction is different from the phantom image. The last column shows the total running time.

Note that we use a different number of projections for each phantom. The results for the two binary phantoms show that there is a sharp lower bound on the number of projections that are required for a reconstruction of high quality. For example, for the *cylinders* phantom, using 9 projections results in a pixel error of 0.83, whereas using 10 or more projections suddenly makes the pixel error drop below 0.002. For each of the phantoms we experimentally determined the minimal number of required projections, indicated by a bold font in Table 1.

The second row of Figure 4 shows ARM reconstruction of the three phantoms, after ten iterations. The number of projections used in Figure 4(d), 4(e), and 4(f) are 5, 10, and 18, respectively. Finally, the last row of Figure 4 shows DART reconstructions of the phantom images. The DART reconstructions are based on the same number of projections as the ARM reconstructions in the middle row.



(a) Simple (b) Cylinders (c) Shepp-Logan

(d) ARM: 5 proj. (e) ARM: 10 proj. (f) ARM: 18 proj.

(g) DART: 5 proj. (h) DART: 10 proj. (i) DART: 18 proj.

Fig. 4. (a-c): original phantom images; (d-f): ARM reconstructions; (g-i): DART reconstructions.

The results show that for each of the phantoms, the DART algorithm is capable of computing an accurate reconstruction from a significantly smaller number of projections than required by the ARM algorithm to obtain similar quality. To the best of our knowledge, there are currently no DT reconstruction results for images larger than 256×256 in the literature.

phantom	d	#DART iters	pixel error	running time (sec.)
Simple	4	120	0.03557	8.7
	5	**110**	**0.00042**	**9.4**
	6	90	0.00026	8.6
Cylinders	9	130	0.08303	18.7
	10	**110**	**0.00175**	**17.4**
	11	120	0.00167	20.5
Shepp-Logan	12	110	0.14213	22.5
	15	60	0.08440	15.3
	18	**100**	**0.02567**	**29.6**
	21	80	0.02355	27.4

Table 1. Experimental results for the three phantoms, using perfect projection data

Even for the phantom images of size 512×512, the DART algorithm required less than half a minute computation time in all cases. An interesting question, which we consider to be out of the scope of this paper, is how the minimal number of required projections can be determined a priori.

4. CONCLUSIONS

We have presented a new iterative algebraic reconstruction algorithm for discrete tomography, called DART. The DART algorithm combines the efficiency of iterative algebraic methods from continuous tomography with the power of discrete tomography to compute accurate reconstructions from relatively few projections.

Our experimental results demonstrate that the DART algorithm is capable of computing reconstructions of very high quality from a small number of projections. The algorithm is very effective for binary images, but it can also be used to reconstruct images that contain more than two gray levels.

5. REFERENCES

[1] G.T. Herman and A. Kuba, Eds., *Discrete Tomography: Foundations, Algorithms and Applications*, Birkhäuser, Boston, 1999.

[2] K.J. Batenburg, "A network flow algorithm for binary image reconstruction from few projections," *Lecture Notes Comp. Sci.*, vol. 4245, pp. 86–97, 2006.

[3] S. Weber, A. Nagy, Th. Schüle, C. Schnörr, and A. Kuba, "A benchmark evaluation of large-scale optimization approaches to binary tomography," *Lecture Notes Comp. Sci.*, vol. 4245, pp. 146–156, 2006.

[4] G.T. Herman and A. Kuba, Eds., *Advances in Discrete Tomography and its Applications*, Birkhäuser, Boston, 2007.

[5] A.C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*, SIAM, 2001.